What is claimed is:

1. A high speed, low power, cacheless computer system comprising:

at least one central processing unit (CPU) to execute instruction fetched by the central processing unit in response to the contents of the instruction selectively fetched from a computer system program;

a main execution memory means comprising a plurality of low cost low power high speed memory banks generally having a slower access time and serving as one of the areas to store said computer system program;

a transition buffer means comprising high speed memory which stores starting locations of program and data segments characterized by a branch or jump instructions and recently used memory references and data values;

an address controller means which includes a circuit to generate and create address for the main execution memory means to access remainder of the computer system program started and stored in said transition buffer means;

a first bus including an address bus able to transport and communicate address information;

a second bus including a data bus able to transport data information;

a third bus including a control bus able to transport and communicate control information; and

said central processing unit, main execution memory means, transition buffer means and address controller means are connected in such a manner that said first bus, second bus and third bus are able to transport and communicate address, data and control information between said central processing unit, main execution memory means, transition buffer means and address controller means; wherein

said central processing unit selectively couples with main execution memory means and executes instructions from the main execution memory means at the start of the program;

said central processing unit selectively couples with the transition buffer means and executes instructions from transition buffer means in such a manner that when a jump or branch instructions are encountered, the central processing unit fetches the starting memory locations of new program branch instruction from the said transition buffer means and continues to execute the instructions;

said central processing unit further couples with said address controller means and said main execution memory means while said central processing unit executes starting locations of a branch or jump location from said transition buffer means and allows enough time to elapse for the main execution memory to access memory locations located on said main execution memory means and place at the output of said main execution memory for the central processing unit to fetch;

said central processing unit then starts executing instructions from the main execution memory means until the central processing unit encounters program change characterized by completion of the program and a new program branch leading to the program locations on the transition buffer means.

2. A high speed, low power, cacheless computer system of claim 1 wherein the transition buffer means is made of static random access memory and said main execution memory means is made of dynamic random access memory.

3. A high speed, low power, cacheless computer system of claim 1 wherein said main execution memory means is made of dynamic random access memory and includes a plurality of groups of parallel memory cells wherein, a group of parallel memory cell is providing program instructions to central processing unit while other group is accessing memory cells for the next group of program instructions.

4. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit includes a logic which allow a group of recently used data to be stored in an area of said of a transition buffer means.

5. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit includes a logic for implementing non pipelined architecture whereby no instruction queue is provided in the central processing unit and includes a memory area which allow a group of recently used data to be stored in a separate area of said transition buffer means.

6. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit includes a logic for implementing a pipelined architecture for implementing program branch instructions whereby at least one instruction queue is provided for implementing program branch instructions in pipeline mode in locations characterized by the central processing unit and transition buffer means and main execution memory means and further includes a memory area which allows a group of recently used data to be stored in a separate area of said transition buffer means.

7. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit includes a logic for implementing a pipelined memory storage in main execution memory means wherein sequential starting locations of multiple branch instructions are stored for implementing program branch instructions as they occur in program execution in required order and determined at the compile time, whereby the starting locations for a new branch will be available to said central processing unit from this pipelined storage architecture.

8.Said pipelined memory storage as claimed in claim 7 has capability to locate in multiple areas characterized by said transition buffer or said main execution memory.

9. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit(CPU ) includes a logic to prevent simultaneous writing of same location by more than one device.

10. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit(CPU ) includes a logic to decode at least one block of a sequence of instruction in advance and make it available for execution when required.

11. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit(CPU ) includes a logic and storage area to store data that can be available to central processing unit (CPU ) at any time and can be related to main execution memory.

12. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit(CPU ) includes a logic and storage area to store data that can be selectively loaded from the main execution memory to the storage area and selectively removed from the storage area to be determined by the execution requirement of the program and conditions determined at the compile time.

13. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit(CPU ) includes a look ahead and decode logic which comprises plurality of logic blocks to decode branch instructions in advance by determining the sequence of branch and jump instructions from the previously stored branch and jump instructions storage area and sequential order determined at the compile time and decode them in advance permitted by program execution conditions, said instruction storage area is located in said main execution memory means or transition buffer means.

14. A high speed, low power, cacheless computer system of claim 1 which includes a Direct access memory means, which further includes logic to transfer data from external device

characterized by a hard drive, said logic allows a small block of data and instructions to be loaded in said transition buffer means and said main execution memory means to start execution of instructions immediately and load remainder of data and instructions as the program starts executing thereby providing fast response direct memory access to said computer system.

15. A high speed, low power, cacheless computer system of claim 1 wherein said central processing unit comprises a logic to implement zero latency interrupt response system that activates interrupt service routine without any delay involved in accessing the main execution memory means by using transition buffer means to start the interrupt service routine.

16. A high speed, low power, cacheless computer system of claim 1 wherein a power management and control logic optimizes power consumption by turning on only the active areas of the system and turning off the remainder of the system components.

17. A high speed, low power, cacheless computer system able to implement Harvard type architecture comprising:

at least one central processing unit (CPU) to execute instruction fetched by the CPU in response to the contents of the instruction and data;

a main execution memory means comprising a plurality of low cost low power high speed memory banks to store instruction program contents for central processing unit (CPU) to execute

instructions and selected data segments for data storage in separate areas and generally having a slower access time;

a main data memory means comprising a plurality of low cost low power high speed memory banks to store data segments for data storage in separate areas and generally having a slower access time;

a transition buffer means comprising high speed memory which stores starting locations for program branch instructions and data segments for data storage in separate memory areas and recently used memory references in selected areas;

an address controller means which includes a means to generate and create address for the main execution memory means for instruction and main data memory for data to facilitate the access to remainder of the instruction and data segments started and stored in the said transition buffer means;

a first bus including an address bus able to transport and communicate address information;

a second bus including a data bus able to transport data information;

a third bus including a control bus able to transport and communicate control information; and

said central processing unit, main execution memory means, transition buffer means and address controller means are connected in such a manner that said first bus, second bus and third bus are able to transport and communicate address, data and control information between said central processing unit, main execution memory means, transition buffer means and address controller means; wherein

said central processing unit selectively couples with main execution memory means and executes instructions from the main execution memory means at the start of the program;

said central processing unit selectively couples with the transition buffer means and executes instructions from transition buffer means in such a manner that when a jump or branch instructions are encountered, the central processing unit fetches the starting memory locations of new program branch instruction from the said transition buffer means and continues to execute the instructions;

said central processing unit further couples with said address controller means and said main execution memory means while said central processing unit executes starting locations of a branch or jump location from said transition buffer means and allows enough time to elapse for the main execution memory means to access memory locations located on said main execution memory means and place at the output of said main execution memory for the central processing unit to fetch;

said central processing unit selectively couples with the transition buffer means and extracts data from the main data memory means and transition buffer means in such a manner that when a data element is needed, the central processing unit fetches the starting memory locations of new data from the said transition buffer means and allows enough time to elapse for the main data memory means to access data locations and place at the output for the central processing unit to fetch, said central processing unit then starts extracting data from the main data memory means until the central processing unit encounters request for new data,

said central processing unit selectively couples with the transition buffer means and writes data to the main data memory means and transition buffer means in such a manner that when a data element is written, the central processing unit writes starting memory locations of new data to the said transition buffer means and allows enough time to elapse for the main data memory means to access data locations and enable the main data memory means for central processing unit to write, said central processing unit then starts writing data to the main data memory means until the central processing unit encounters request for new data to be written.

said central processing unit then starts executing instructions from the main execution memory means until the central processing unit encounters program change characterized by completion of the program and a new program branch leading to the program locations on the transition buffer means.

18. A high speed, low power, cacheless computer system of claim 17 wherein said central processing unit(CPU ) includes a logic and storage area to store data that can be available to central processing unit (CPU ) at any time and can be related to main data memory said central processing unit(CPU ) further includes a logic to prevent simultaneous writing of same location by more than one device.

.

19. A high speed, low power, cacheless computer system comprising:

plurality of central processing units (CPU) to execute instruction fetched by each of the central processing unit in response to the contents of the instruction selectively fetched from a computer system program;

a main execution memory means comprising a plurality of low cost low power high speed memory banks generally having a slower access time and serving as one of the areas to store said computer system program ;

a transition buffer means comprising high speed memory which stores starting locations of program and data segments characterized by a branch or jump instructions and recently used memory references and data values;

an address controller means which includes a circuit to generate and create address for the main execution memory means to access remainder of the computer system program started and stored in said transition buffer means;

a first bus including an address bus able to transport and communicate address information;

a second bus including a data bus able to transport data information;

a third bus including a control bus able to transport and communicate control information; and

said central processing unit, main execution memory means, transition buffer means and address controller means are connected in such a manner that said first bus, second bus and third bus are able to transport and communicate address, data and control information between said central processing unit, main execution memory means, transition buffer means and address controller means; wherein

at least one central processing unit selectively couples with main execution memory means and executes instructions from the main execution memory means at the start of the program;

at least one central processing unit selectively couples with the transition buffer means and executes instructions from transition buffer means in such a manner that when a jump or branch instructions are encountered, the central processing unit fetches the starting memory locations of

new program branch instruction from the said transition buffer means and continues to execute the instructions;

said at least one central processing unit further couples with said address controller means and said main execution memory means while said central processing unit executes starting locations of a branch or jump location from said transition buffer means and allows enough time to elapse for the main execution memory to access memory locations located on said main execution memory means and place at the output of said main execution memory for the central processing unit to fetch;

said central processing unit then starts executing instructions from the main execution memory means until the central processing unit encounters program change characterized by completion of the program and a new program branch leading to the program locations on the transition buffer means.

20. A high speed, low power, cacheless computer system of claim 19 wherein said central processing unit(CPU ) includes a logic to prevent simultaneous writing of same location by more than one device.